

3-D

SUPERGRAPHICS

AND COLOR GAME DEVELOPMENT SYSTEM

For Apple II or Plus

IN COLOR

USA UNITED
SOFTWARE
OF
AMERICA



Supergraphics Three-Dimensional Display System

P. Lutus, Kerby, Oregon 1980

Contents

Page 1: Introduction

Pages 2 - 10: Part I: Basic Supergraphics

Source of Images - Cartesian Coordinates - Command Numbers - Color Entries - Coordinate Numbers - BASIC DATA table format - Enabling Supergraphics - Image Table Identification - Example Images - Display Commands - Scale - Rotation - Translation - Position - Erase - Text Color - Quit

Pages 11 - 15: Part II: Advanced Supergraphics

Command Loops - Multiple Commands - Multiple Images - Value Query - Text Cursor Display - VTAB, HTAB from BASIC

Pages 16 - 18: Part III: Use of the GRAFBASIC utility

Menu - Draw - Entry Format - Show - Show Commands - File

Pages 19 - 21: Part IV: System Technical Description

Display method - Screen Sequence - Procedure - Matrices - Plot format

Page 22: Appendix A: Short-form list of Supergraphics Commands

Page 24: Appendix B: System Memory Map

Page 25: Appendix C: Notes to Atari Version

Page 26: Appendix D: DEMO Program BASIC Listing

Page 31: Appendix E: GRAFBASIC Program BASIC Listing

Supergraphics Three-Dimensional Display System

P. Lutus Kerby, Oregon 1980

Introduction

Supergraphics is a software system that projects three-dimensional color images on the Apple II screen. The system draws images so fast that animated displays are possible. The program is completely compatible with Integer and Applesoft BASIC languages, the Disk Operating System, and other assembly-language programs. Required are an Apple II with 48K of Random Access Memory and a minimum of 1 disk drive.

The quality of the graphic images is much better than that normally available from Apple. All high-resolution graphics colors are available, both for image drawing and text displays.

Access from BASIC is very simple. After initialization, simple printed commands such as are used by the Disk Operating System command the drawing, rotation, scale etc. of the images.

Demonstration and image development programs are provided to aid the software developer. Because Supergraphics is a software development tool, these instructions assume minimum familiarity with the BASIC programming language and the Apple II.

In these instructions, segments of particular interest are preceded by ">>>" in the left margin.

Part I: Basic Supergraphics

Your first step in getting to know Supergraphics should be the demonstration program "DEMO" provided on the system diskette (a listing of "DEMO" is included in these instructions). This demonstration shows the main features of the program and some typical displays. If you are familiar with Apple II, you will take note of the fact that text and high-resolution images are combined, one of the features of Supergraphics.

>>> Source of images

NOTE: The system diskette program "GRAFBASIC", described in Part III, will create images for you using a display and text editor format. The following section describes the image format for the benefit of programmers. It is included for reference, and may contain more information than needed by normal users.

One of the principal purposes of Supergraphics is to project three-dimensional images. The source for these images is a table of coordinate numbers that define lines to be drawn.

>>> Cartesian Coordinate System

For each line ending in the image, three coordinate numbers are required, one for each dimension of space. The three numbers are assigned the letters X, Y, and Z.

X (by convention) defines a point along the horizontal axis. Y defines the vertical axis and Z defines a point in depth. This system of numbers is called the "Cartesian Coordinate System", developed by Renee Descartes.

Imagine that you are in a large city. Your friend is trying to explain the route to his house. He lives three blocks east and seven blocks north of

the bus station and, finally, on the eleventh floor of his building. If he and you knew the Cartesian Coordinate System, he might say "Walk along X until it is equal to 3. Now walk along Z until it is equal to 7. Now move along Y until it is equal to 11. Knock on my door".

Supergraphics accepts numbers in just this way. A table of numbers is created, describing a house or an airplane or anything else. Then Supergraphics is told where the numbers are located in Apple's memory.

>>> Allowable Image number range

The value range of the numbers that will be accepted by Supergraphics is (integer) -128 to 127 in the three dimensions X, Y, and Z. This means that there are about 16 million definable points in the mathematical space of Supergraphics.

Image coordinate numbers are in sets of four. The first number is a command that carries information about color and whether to draw a line or simply move. The remaining three numbers are the X, Y, and Z Cartesian coordinates. Here is a description of the number set:

>>> Command Number Format

Number 1: Command. If this number is less than 128, Supergraphics will move, not draw a line, to the coordinates that follow. If the number is 128 or greater, a line will be drawn. If the number equals 255, this signals Supergraphics that the end of the image table has been reached.

>>> Color Entries

A color number may be added to the command number if desired. Here are the colors and their numbers:

Number Color

1	Green
2	Violet
3	White
4	Black
5	Orange
6	Blue
7	White

These colors and their behavior are described in Apple literature, particularly the "Applesoft Tutorial" and "Apple II Reference Manual".

To create a command number, select a color if desired (this should only be done if the color must be changed), then add 128 to the color number if a line is to be drawn. If there is no color number added to the command value 0/128, the previous color will be used. Examples:

Number	Means
--------	-------

131	Draw a line using color 3 (131=128+3)
-----	---------------------------------------

5	Change to color 5 and move to the given coordinates
---	---

128	Draw a line using the previously entered color
-----	--

>>> Coordinate Number Format

Numbers 2 - 4: These entry numbers are the X, Y, and Z coordinates for a point. They are interpreted by Supergraphics as lying in the range -128 to 127, although BASIC will not place negative numbers directly into memory. To prepare the numbers for placement into memory, simply add 256 to those numbers that are less than 0. A BASIC example:

```
100 N = N + 256 * (N < 0)
```

```
110 POKE A , N
```

>>> BASIC DATA Table format

A simple way to make an image available (not as simple as using the program "GRAFBASIC") is to enter the numbers into BASIC DATA statements, then use a loop to place them into memory. Here is an example program to do this:

```
10 FOR ADDRESS = START TO HIGHNUMBER
20 READ NUMBER
30 IF NUMBER = 255 THEN 70
40 NUMBER = NUMBER + 256 * (NUMBER < 0)
50 POKE ADDRESS,NUMBER
60 NEXT ADDRESS
70 POKE ADDRESS,NUMBER
80 REM THE REST OF THE PROGRAM
,
,
,
999 REM IMAGE DATA STATEMENTS
1000 DATA 3 , -127, -127, 0
1010 DATA 128, 127, -127, 0
1020 DATA 128, 127, 127, 0
1030 DATA 128, -127, 127, 0
1040 DATA 128, -127, -127, 0
1050 DATA 255
```

The DATA table provided will draw a square in the X, Y plane.

>>> Enabling Supergraphics

To enable Supergraphics, first load it into memory:

```
100 PRINT D$ ; "BLOAD OBJ.SUPERG":REM D$ CONTAINS
CONTROL-D
```

Then use this command to enable it:

```
110 CALL 25472
```

After this "CALL" command has been issued (from the keyboard or from a program), Supergraphics will assume control of all display functions on Apple II. All text printing and graphics display will be performed by Supergraphics.

NOTE: There is no need to include the normal graphics command "HGR" in your program. No features of Apple Graphic firmware are used by Supergraphics.

>>> Image Table Identification

At this point, Supergraphics will print any text provided, but it won't display any graphic images. To make Supergraphics display an image, we must tell it where the image is located in memory. Let's assume that an image has been loaded into memory starting at decimal address 4096. The command to enable graphics is:

```
120 PRINT "%SH 4096"
```

Remember: Each command to Supergraphics must begin with a "%" percent sign. The command "%SH" means "shape" and it is followed by the beginning address of the image table. After this command has been printed by BASIC (like all Supergraphics commands, it won't appear on the screen), the image will appear.

HINT: For experimentation purposes, you may issue these commands from the keyboard rather than as part of a program. Example:

```
PRINT "%SH 4096"
```

>>> Example Images

To speed up the learning process, you may want to load the demonstration images from the system diskette rather than create your own. To do so, type:

```
BLOAD IMAGES
```

Now the following images will be accessed by the "%SH" addresses listed:

Command	Image
"%SH 28672"	Spaceship
"%SH 29189"	Complex Spaceship
"%SH 28736"	"APPLE" sign
"%SH 28864"	X, Y, Z Coordinate arrows
"%SH 28928"	Hang glider
"%SH 29017"	Butterfly (wings up)
"%SH 29089"	Butterfly (wings down)
"%SH 29389"	Trapezoid

>>> Display Commands

NOTE: Be sure to see Appendix A: Short-form listing of Supergraphics Commands.

Proper Supergraphic command format within a BASIC program requires (as for DOS) that a carriage return precede the command line. Here are examples:

This example will not work:

```
10 PRINT "HELLO"; : REM NO CARRIAGE RETURN
20 PRINT "%RX 20"
```

This example will work:

```
10 PRINT
20 PRINT "%RX 20"
```

It is not usually necessary to PRINT a carriage return, as in the example above. If programming problems arise, or a Supergraphics command is printed instead of being executed, this may be the problem.

NOTE: Use of the "%" sign in BASIC printing is permitted, in positions other than at the line beginning. Example:

```
10 PRINT "100 % SCORE!"
```

The numeric entries to these commands may be in the range (+,-) 65536. The entries should be integers.

Floating point numbers will cause difficulties.
Example:

```
100 X = INT (X)
110 PRINT "%RX ";X
```

>>> Display Scale

Scale numbers in the range -31 to +31 may be entered to control the overall image size. Negative scale numbers will create an inverted image.
Example:

```
"%SC 20" (SCale)
```

Scale numbers may also be used to specifically control the scale of the X and Y dimensions. These commands stretch the image in the specified dimension. Examples:

```
"%SX 20" (Scale X)
```

```
"%SY -5"
```

>>> X Y, Z Rotation

Here are the Supergraphics commands to rotate the image about the axes X, Y, and Z:

```
"%RX 10" (Rotate X)
```

```
"%RY 20"
```

```
"%RZ -15"
```

The number values are in angular units. Each angular unit is equal to 1.4 degrees (360/256).

>>> Translation

"Translation" means movement along the X, Y, and Z axes. If the image is large, the range of translation will be correspondingly smaller, because the limits of the mathematical space are (+,-) 127 in each dimension. If the image values plus the translation value exceed this range, the

image will "wrap around" within the defined space.
Example:

```
"%TX 20" (Translate X)
"%TY -50"
"%TZ 12"
```

>>> Position

This command displaces the image on the Apple II graphics screen. Normally the center of the image is at $X = 128$, $Y = 96$. This value is modified by the entered number. Example:

```
"%PX -30" moves the X screen image position from
128 to 98
"%PY 20" moves the Y screen image position from 96
to 116
```

NOTE: Unlike Apple II graphics, Supergraphics obeys the mathematical convention that the lower left of the screen is $X = 0$, $Y = 0$.

>>> Default Graphics Color

If an image file has no color specifications, it will be drawn in the default color (3). This default color may itself be changed using this command:

```
"%CG 5" (Color Graphic = 5)
```

This color will be used until (if) a color specification appears in the image file. This makes it possible to change the color of an image in sequential drawings.

>>> Erase screen

This command erases the text and graphics display. There is no numeric entry. Example:

```
"%NW" (New)
```

>>> Enable, disable drawing

These commands enable and disable graphic image drawing, while permitting normal text displays on the graphic screen. Examples:

"%ND" (No Draw)

"%YD" (Yes Draw)

>>> Select Graphic, Text

These commands select the Apple II graphic image or the normal text display. Examples:

"%TE" (Text)

"%GR" (Graphics)

>>> Text Color

This command selects the color in which text will be printed. Example:

"%CT 5" (Color Text = 5)

NOTE: Because of the way by which Apple II generates colors, there will be some distortion of the printed text when colors other than whites 3 and 7 are used. Special effects may be obtained through experimentation. Color 3 is the normal (default) text color. Color 7 gives a "Bold face" appearance to the text, useful for emphasis.

NOTE: To print inverted text (black on white), simply enter the Applesoft BASIC command "INVERSE" (in Integer BASIC, "POKE 50, 63").

>>> Quit Supergraphics

To disable Supergraphics and return to normal Apple II operation, use this command:

"%QT" (Quit)

Part II: Advanced Supergraphics

This section describes advanced instructions, combinations of instructions and methods.

>>> Command Loops

In a BASIC program one would rotate an image this way:

```
10 FOR A = 0 TO 256 STEP 4
20 PRINT "%RX" ; A
30 NEXT X
```

This method has the significant drawback that Applesoft BASIC is more than 100 times slower than Supergraphics. Therefore Supergraphics has its own control loop capability. Here is an example that duplicates the function of the above BASIC segment:

```
10 PRINT "%RX 0 T 256 S 4"
```

NOTE: When using the loop capability, remember that, unlike BASIC, the controlled value must equal the ending value at termination. This example:

```
10 PRINT "%RX 0 T 255 S 4"
```

will loop endlessly, since the controlled value will not equal 256 during operation.

Negative numbers may be used in loop commands:

```
10 PRINT "%RX 0 T -256 S -4"
```

NOTE: Pressing any keyboard key will exit a loop command. This is a useful feature in game programs, in which constant animated displays may be controlled by the keyboard. This means that a loop can be created to rotate an image while a keyboard entry is awaited. Example:

```
10 PRINT "%RX 0 T 255 S 4" : REM ENDLESS LOOP
```

```
20 INPUT N$
```

>>> Multiple command lines

Each time a command line is printed, the image is updated. If it is desired to change more than one display constant before displaying the image, simply enter all values to be changed on a single line. Example:

```
10 PRINT "%RX ";X;" ,RY ";Y;" ,RZ";Z
```

Each command is separated in the line by commas. Multiple loops may be entered:

```
10 PRINT "%RX 0 T 256 S 4,RY 0 T 256 S 4,RZ 0 T 256  
S 4"
```

These loops will be executed sequentially.

>>> Multiple images

NOTE: To fully understand the following method, particularly the reference to two display screens, read Part IV: System Technical Description.

Here is the method to draw multiple images or copies of an image. Normally Supergraphics "undraws" the previous image before drawing the new image or image position. To prevent this erasure, simply make a "%SH" shape address entry. The "%SH" command abandons the "undraw" data normally used to erase the previous image. Example:

```
10 FOR X = -64 TO 64 STEP 32  
20 PRINT "%SH 4096,PX";X  
30 PRINT "%": REM PLACE IMAGE ON BOTH SCREENS  
40 NEXT X
```

Notice the null command "%" in line 30. This command simply redraws the image without changing anything, assuring that both screens carry a copy of the image. The image would otherwise flash as Supergraphics switches between screens.

This example makes five copies of the image on the screen. The described method may be used to repeat an image, or develop many images without erasing the old ones.

NOTE: If difficulties arise while using a combination of rotation and image switching, simply be sure that both screens carry the same image (see line 30 in the above example).

To return to a previous image position, use the above method, specifying the old image position (or rotation, scale etc.), and make two copies of the drawing before moving. Example:

```
10 PRINT "%SH 4096,PX"; OLDX ;",PY"; OLDY
20 PRINT "%"
30 REM OK TO MOVE
```

>>> Images with moving parts

There is a variation on the method just described. In this variation, the shape table address is changed, but the "undraw" information is retained. This means that you can flap the wings on a butterfly, for example, or other things that require two or more versions of an image.

In the "DEMO" program this technique is used to display the butterfly image. Two butterfly images, one with wings up, the other with wings down, are used. So it is necessary to change the image address, but still "undraw" the previous images. Here is an example:

```
10 FOR Q = 0 TO 30
20 PRINT "%SH + ";WINGSUP
30 PRINT "%SH + ";WINGSDOWN
40 NEXTQ
```

The address numbers "WINGSUP" and "WINGSDOWN" point to the beginning addresses of the two versions of the butterfly image.

Notice the "+" sign between "%SH" and the number. This "+" sign, when used with the "%SH" command, preserves the "undraw" information. If the "+" sign is not used, the undraw information is abandoned, as in the previous example.

>>> Getting a value from Supergraphics

It may be necessary to obtain a program value from Supergraphics, rather than transmitting one. To do this, use the following procedure:

```
10 PRINT "%RX ?"  
20 RX = PEEK (6)
```

The requested command value will be placed in memory location 6, where it can be recovered with the BASIC "PEEK" feature. In the case of the query:

```
10 PRINT "%SH ?"
```

The present shape address (two bytes) will be provided in locations 6 and 7. To obtain the address value, use this method:

```
20 ADDRESS = PEEK (6) + PEEK (7) * 256
```

>>> Text Cursor Display

One difference between Supergraphics and the ordinary Apple text display is that a cursor won't automatically appear when a line input is being accepted. To make a cursor appear, simply print a backspace before the BASIC INPUT statement:

```
10 PRINT "ENTER YOUR NAME : " ; CHR$(8);  
20 INPUT "" ; N$
```

The Supergraphics cursor will behave in much the same way as the normal Apple cursor, with the exception of the ESC cursor control keys. To print a cursor in any circumstances (not necessarily for an INPUT statement), print it as a character:

10 PRINT CHR\$(95)

>>> Text Clear

The BASIC command "HOME" won't clear the Supergraphics screen. The previously described "%NW" command should be used to clear the Supergraphics display and resets the cursor position to the upper left.

>>> VTAB, HTAB, display window

These BASIC features are duplicated in Supergraphics. The Apple II window control locations (decimal 32,33,34,35) and the tab control commands all function normally. The text display will scroll correctly (somewhat more slowly than normal), but this may cause difficulties if an image is being updated, since relative screen positions will be modified by the scroll.

Part III: Use of the "GRAFBASIC" utility program

This program may be used to create, view and save Supergraphic images for use elsewhere. To operate the program, type "RUN GRAFBASIC" (Applesoft BASIC is required). A menu will appear.

The menu options permit you to "(S)how", "(D)raw", and "(F)ile" an image. First, let's review "(D)raw". Press "D".

A display of the image (if any) and the last 6 lines of the image table will be displayed, followed by a prompt:

(D)RAW, (M)OVE, (1-7) COLOR, (-) DELETE

EXAMPLES: 'D3', 'M'. ENTER:

The "-" command will delete the last entry. The "D"raw command will command that a line be drawn. The "M" command moves without drawing. These commands may be followed by a color number (1-7). Remember that a color number should only be entered if a color change is required. Specifying a color for each line slows the display.

After entering "D" or "M" and the optional color number, a new prompt will appear:

'D' X, Y, Z:

Now enter the three numbers for X, Y, and Z. The numbers should be integers in the range -127, +127. After the entry, press RETURN. The entered line will appear in the data table and (if command "D" has been used) the line will appear in the graphic display. To quit the "(D)raw" feature, simply press RETURN with no entry from the first of the "(D)raw" prompts.

NOTE: For maximum display speed, make images as much as possible out of line continuations rather than moves. This is because each line ending must

be calculated, and calculation time is a significant percentage of display time. Another speed hint is to only specify color in line entries when color is to be changed.

To "(S)how" the image, press "S" from the menu. There are a number of options available to rotate, scale, and otherwise modify the image.

To "R"otate, "T"ranslate, "P"osition, or "S"cale the image, simply press the corresponding letter key. Now press "X", "Y", or "Z" to carry out the command on these dimensions. To change commands, press one of the letters "R,T,P,S" again.

To change the direction of movement while pressing the "X,Y,Z" keys, press "-". Subsequent "X,Y,Z" keys will move in the reverse direction. To change the step size, press "I" and enter the new increment value.

To zero all display constants, press "*". Rotation, scale, translation and other values will be set to 0 or their default values.

To make a direct Supergraphics "C"ommand, press "C". Example:

ENTER SUPERGRAPHICS COMMAND: RX 0 T 256 S 4

NOTE: Don't enter the "%" command token before your command in this feature. The token is added automatically.

NOTE: All rotations and color inputs made to "(S)how" are preserved for use elsewhere in GRAFBASIC. Scale returns to the default value of 8.

To quit the "(S)how" feature, press RETURN with no command entry.

To "(F)ile" your image, press "F" from the menu. A file name will be requested, followed by a specification of "(L)oad or (S)ave". The most

recently accessed disk drive will yield or receive the image file.

Part IV: System Technical Description

Supergraphics uses both Apple II high resolution display screens for image display. This produces a flicker-free graphic display, because image erasure and redrawing is conducted on the invisible screen. Therefore it is necessary to make two copies of an image that is to remain on display. Example:

```
10 PRINT "%RX -32" : REM ROTATE X AXIS
20 PRINT "%"       : REM COPY TO OTHER SCREEN
```

This procedure need only be used if another image is to be drawn and it is desired to keep the first on display.

Text display is made to both screens simultaneously, so that graphic screen switching won't disrupt the text display.

Supergraphics uses the following general procedure for each image update:

1. Prepare access to the screen not presently on display.
2. Undraw the old version of the image, if present, by zeroing all bytes that were used for the image.
3. Compute new three-dimensional matrices (see below).
4. Draw updated image, using matrix transformation.
5. Switch display to the new image screen.

NOTE: An additional version of Supergraphics is available on the system diskette. Named SUPERB&W, this version draws only in black and white. It may be used in exactly the same way as described for normal Supergraphics. Because it draws in black and white, the lines drawn are somewhat more precise. Also, this version draws more quickly. Therefore if a black and white monitor is in use, or maximum speed is desired, use SUPERB&W.

Supergraphics uses a fast assembly-language matrix

multiplication algorithm to provide the mathematical transformations needed for three-dimensional display. The following 3 by 3 matrices are multiplied together to provide the display matrix:

(X),(Y),(Z) = rotation angles in units (1 unit = 1.4 degrees)

X,Y,Z = Cartesian Scalar values

Matrix 1:

Scale X	0	0
0	Scale Y	0
0	0	Scale Z

Matrix 2:

1	0	0
0	cos (X)	-sin (X)
0	sin (X)	cos (X)

Matrix 3:

cos (Y)	0	-sin (Y)
0	1	0
sin (Y)	0	cos (Y)

Matrix 4:

cos (Z)	-sin (Z)	0
sin (Z)	cos (Z)	0
0	0	1

The preceding matrices are computed once per image.

The result matrix of the previous multiplications is then used to transform each line ending in the image, using this final matrix:

X+X Translation Y+Y translation Z+Z Translation

0 0 0

0 0 0

The plotting coordinates X' and Y' are then offset by the coordinates of the screen center:

Plot $X = X' + 128 + X \text{ Position}$

Plot $Y = Y' - 96 + Y \text{ Position}$

Appendix A: Short-form list of Supergraphics commands

The commands listed below become operative after Supergraphics has been loaded and initialized. The initialization is by way of the command "CALL 25472". Here is a BASIC program example:

```
10 CALL 25472
```

All entries are decimal integers in the range +-65536. Some of the commands use only the subset +-256 of the entered numbers.

Most of these commands may be used in the loop entry:

```
"%(command) (number) T (number) S (number)"
```

This loop entry is equivalent to the basic statement:

```
FOR (number) TO (number) STEP (number)
```

Command	Means	Default Value	Comment
"RX"	Rotate	X 0	in units of angle
"RY"	Rotate	Y 0	1 unit = 1.4 degrees
"RZ"	Rotate	Z 0	(64 = 90 degrees)
"TX"	Translate	X 0	limited to bounds
"TY"	Translate	Y 0	of +-127 space
"TZ"	Translate	Z 0	including image values
"PX"	Position	X 0	added to 128
"PY"	Position	Y 0	added to 96
"SX"	Scale	X 8	range +-31
"SY"	Scale	Y 8	
"SZ"	Scale	Z 8	
"SC"	Scale X,Y,Z	8	

"CG"	Color Graphics 3 range 1-7
"CT"	Color Text 3
"GR"	Enable Graphics
"TE"	Enable Text
"YD"	Yes Draw
"ND"	No Draw
"NW"	New (erase)
"QT"	Quit
"SH"	Shape (address) Abandons "undraw" data unless followed by "+"

Appendix B: Supergraphics Memory Map

\$0000 - \$00FF	6502 Page Zero
\$0100 - \$01FF	6502 Stack
\$0200 - \$02D1	BASIC Input Buffer
\$02D2 - \$03CF	Supergraphics Variable Locations
\$03D0 - \$3FF	DOS Links
\$0400 - \$07FF	Text Display Page
\$0800 - \$1FFF	Applesoft BASIC Program Area
\$2000 - \$5FFF	Supergraphics Display Area
\$6000 - \$637F	Lookup Tables
\$6380 - \$6FFF	Supergraphics Program
\$7000 - \$95FF	Applesoft String Variables
\$9600 - \$CFFF	Disk Operating System

Appendix C: Notes to Supergraphics Atari Version

All effort has been expended to make the Apple II and Atari 800 Supergraphics versions as much alike as possible. The following procedures should be used on the Atari version only:

1. Loading Supergraphics

Either:

A. From Disk Utility, type "L", then "SUPERG"

Or:

B. BASIC program load example:

```
10 OPEN #1,4,0,"D:SUPERG"
20 TRAP 70
30 FOR A = 22010 TO 32000
40 GET #1,X
50 POKE (A) ,X
60 NEXT A
70 CLOSE #1
80 GRAPHICS 24
90 X = USR (22016)
100 OPEN #1,12,0,"G:"
110 PRINT #1,"HELLO SUPERGRAPHICS!"
```

Notice that the starting address for the loop is the file beginning - 6. This is because Atari has a special header on binary load files that must be discarded. All printing in the program MUST be to the I/O block (in this example, #1) opened for Supergraphics, otherwise there will be a asty exit from the Supergraphics features.

The Atari version of Supergraphics lies in memory between hex \$5600 - \$6150. Two graphics screens are used, lying between \$6150 - \$A000.

There are two extra commands in the Atari version, used to control screen print position. Example:

```
10 PRINT #1, "%VT10" : REM LINE 10
20 PRINT #1, "%HT15" : REM COLUMN 15
```

Appendix D: DEMO Program BASIC Listing

```
10 GOTO 230
20 REM
30 REM DRAWS FIGHTERS
40 REM
50 PRINT "%ND,SH28672,SC0,CG";C;",";RY"; - 20 *
S;",";YD"
60 FOR X = 0 TO 20 * T STEP 2 * T
70 PRINT "%SC";X;",";RZ";X * 12;",";PX";X * 5 *
S;",";PY"; - X
80 NEXT X
90 PRINT "%": PRINT "%": PRINT "%"
100 RETURN
110 REM
120 REM KEYIN ROUTINES
130 REM
140 KF = 128
150 POKE - 16368,0
160 VTAB 24: PRINT " ( PRESS ANY KEY TO CONTINUE
) "; CHR$(8);
170 FOR Z = 0 TO WT
180 IF PEEK (- 16384) + KF < 128 THEN NEXT Z
190 KF = 0: VTAB 23: HTAB 1: PRINT : VTAB 1: HTAB
1: RETURN
200 REM
210 REM ENTRY POINT: SG LOADED?
220 REM
230 PRINT "BLOADOBJ.SUPERG"
240 PRINT "BLOADIMAGES"
250 REM
260 REM FIRST DISPLAY
270 REM
280 CLEAR
290 CALL 25472:S = 1:C = 1:T = 1
300 GOSUB 140:W = 20
310 VTAB 1: INVERSE : PRINT "%CT";C
320 PRINT " "
330 PRINT " << SUPERGRAPHICS (C) P. LUTUS 1980 >>"
340 PRINT " "
350 PRINT "%CT3": NORMAL : PRINT
360 PRINT "... AN ";: INVERSE : PRINT
"INCREDIBLE";: NORMAL : PRINT " GRAPHICS PROGRAM."
```

```
370 C = C + 1: IF C = 3 THEN C = 5
380 IF C > 7 THEN C = 1
390 S = - S
400 IF S = 1 THEN T = - T
410 GOSUB 50
420 IF PEEK ( - 16384) > 128 THEN 440
430 W = W - 1: IF W > 0 THEN 310
440 REM
450 REM APPLE SIGN DISPLAY
460 REM
470 PRINT "%ND,NW,SH28736,RZ": VTAB 11
480 PRINT "SUPERGRAPHICS IS AN ";
490 INVERSE : PRINT "APPLE ENHANCER";: NORMAL :
PRINT "."
500 PRINT "IT WILL DO THINGS APPLE CAN'T."
510 PRINT : PRINT "SUPERGRAPHICS CAN:": PRINT
520 PRINT "1. DRAW THREE-DIMENSIONAL COLOR
OBJECTS"
530 PRINT "    AT RATES UP TO 30 PER SECOND."
540 PRINT
550 PRINT "%CT5"
560 PRINT "2. DISPLAY 6 COLOR AND ";
570 INVERSE : PRINT "INVERSE";: NORMAL
580 PRINT " TEXT."
590 PRINT "%CT3"
600 PRINT
610 PRINT "3. ALLOW YOU CONTROL FROM ";
620 INVERSE : PRINT "BASIC";: NORMAL
630 PRINT " USING": PRINT "    SIMPLE PRINTED
COMMANDS."
640 GOSUB 140
650 PRINT "%PX,PY52,SC20,RX10,YD,RYOT2048S8"
660 REM
670 REM X,Y,Z FRAME DISPLAY
680 REM
690 PRINT
"%ND,NW,SH28864,RX10,RY10,PX64,PY48,SC20"
700 PRINT "IMAGES ARE PROVIDED TO SUPERGRAPHICS"
710 PRINT "AS SETS OF THREE NUMBERS: X, Y, AND Z."
720 PRINT : PRINT "X (    ) IS HORIZONTAL
POSITION."
730 PRINT "Y (    ) IS VERTICAL POSITION."
740 PRINT "Z (    ) IS DEPTH."
750 PRINT : PRINT "WITH THESE NUMBERS, ANY POINT
```

IN SPACE"

760 PRINT "MAY BE DEFINED.

770 PRINT : PRINT "WHILE DRAWING THE IMAGE,
SUPERGRAPHICS"

780 PRINT "MAY BE COMMANDED TO ROTATE AROUND"

790 PRINT "THE X, Y, OR Z AXES, OR MOVE ALONG THE"

800 PRINT "AXES. THE IMAGE MAY BE STRETCHED ALONG"

810 PRINT "THE AXES AS WELL."

820 GOSUB 140

830 INVERSE : FOR Q = 1 TO 3

840 PRINT "%CT";Q

850 VTAB 3 + Q: HTAB 5: PRINT

860 NEXT Q: NORMAL

870 PRINT "%YD,PX64,PY-58,SC15,RX-20,RY8T4096S8"

880 PRINT "%NW,ND,PX,PY,RX-20,RY-20,SC30"

890 PRINT "HERE ARE EXAMPLES OF SUPERGRAPHICS"

900 PRINT "IMAGE CONTROLS:"

910 VTAB 22: PRINT "X AXIS ROTATION"

920 PRINT "%YD,RX-12T236S8"

930 VTAB 22: PRINT "Y"

940 PRINT "%RY-12T236S8"

950 VTAB 22: PRINT "Z"

960 PRINT "%RZ8T256S8"

970 VTAB 22: PRINT "X AXIS TRANSLATION"

980 PRINT "%TX-32T32S4,TX"

990 VTAB 22: PRINT "Y"

1000 PRINT "%TY-32T32S4,TY"

1010 VTAB 22: PRINT "Z"

1020 PRINT "%TZ-32T32S4,TZ"

1030 VTAB 22: PRINT "X AXIS SCALING

1040 PRINT "%SX30T-30S-3,SX-30T30S3"

1050 VTAB 22: PRINT "Y"

1060 PRINT "%SY30T-30S-3,SY-30T30S3"

1070 WT = 500: GOSUB 150

1080 REM

1090 REM HANG GLIDER DISPLAY

1100 REM

1110 PRINT "%NW,ND,SH28928,SC8"

1120 VTAB 12: PRINT "AMAZING DISPLAYS CAN BE
DEVELOPED USING"

1130 PRINT "THE FEATURES OF SUPERGRAPHICS."

1140 PRINT

1150 PRINT "GAME PROGRAMS CAN BE WRITTEN IN WAYS"

1160 PRINT "NEVER BEFORE POSSIBLE."


```
1170 PRINT : PRINT "REMEMBER: ALL SUPERGRAPHICS
FEATURES"
1180 PRINT "ARE AVAILABLE FROM INTEGER OR
APPLESOFT"
1190 PRINT "BASIC WITH A VERY SIMPLE COMMAND"
1200 PRINT "FORMAT."
1210 GOSUB 140
1220 CLEAR
1230 C = 0:S = 1:W = 8
1240 S = - S
1250 RX = 0:RY = 64 * S:RZ = 32:PX = 0:PY = 48
1260 C = C + 1: IF C = 3 THEN C = 5
1270 IF C > 6 THEN C = 1
1280 PRINT "%CG";C
1290 FOR X = - 92 * S TO 92 * S STEP 6 * S
1300 IF PEEK ( - 16384) > 128 THEN 1420
1310 RX = INT (RX - 5 + ( RND (1) * 10))
1320 RY = INT (RY - 5 + ( RND (1) * 10))
1330 RZ = INT (RZ - 5 + ( RND (1) * 10))
1340 PY = INT (PY - 5 + ( RND (1) * 10))
1350 IF PY < 48 THEN PY = PY + 4
1360 PRINT
"%YD,PX";X;","PY";PY;","RX";RX;","RY";RY;","RZ";RZ
1370 NEXT X
1380 PRINT "%ND":W = W - 1: IF W > 0 THEN 1240
1390 REM
1400 REM BUTTERFLY
1410 REM
1420 PRINT "%ND,NW,SH29017,PY65,RX30,RZ"
1430 CLEAR :CG = 1:S = 1:W = 16
1440 N1 = 1:N3 = 3:N5 = 5:N7 = 7
1450 POKE - 16368,0: VTAB 9
1460 PRINT "SUPERGRAPHICS COMMANDS ARE VERY
SIMPLE."
1470 PRINT : PRINT "EXAMPLE: 10 PRINT "; CHR$
(34);"%RX 10"; CHR$ (34): PRINT
1480 PRINT "THIS COMMAND MEANS 'ROTATE THE X AXIS"
1490 PRINT "BY 10 UNITS'."
1500 PRINT : PRINT "THERE ARE NO HARD-TO-REMEMBER
'PEEKs'"
1510 PRINT "AND 'POKES' IN SUPERGRAPHICS."
1520 PRINT : PRINT "SUPERGRAPHICS IS DESIGNED TO
BE VERY"
1530 PRINT "SIMPLE TO USE."
```

```
1540 GOSUB 140
1550 PRINT "%ND,CG"CG
1560 FOR PX = - 96 * S TO 96 * S STEP 8 * S
1570 RY = PX * N3
1580 PRINT "%YD,SH+29017,PX"PX",RY"RY
1590 IF PEEK ( - 16384) > 128 THEN 1660
1600 PRINT "%SH+29089"
1610 NEXT PX
1620 S = - S
1630 CG = CG + N1: IF CG = N3 THEN CG = N5
1640 IF CG = N7 THEN CG = N1
1650 W = W - 1: IF W > 0 THEN 1550
1660 POKE - 16368,0
1670 A = 29189:S = 8:W = 2: GOSUB 1710
1680 A = 29389:S = 4:W = 6: GOSUB 1710
1690 A = 28928:S = 8:W = 3: GOSUB 1710
1700 GOTO 280
1710 C = 1: PRINT : PRINT "%NW,ND"
1730 VTAB 20: PRINT "ANY NUMBER OF IMAGES CAN BE
ON DISPLAY"
1740 PRINT "UNDER INDIVIDUAL PROGRAM CONTROL.":
GOSUB 140
1750 FOR Y = - 10 TO 48 STEP 58
1760 FOR X = - 64 TO 64 STEP 128
1770 PRINT "%ND,SH";A;","PX";X;","PY";Y;","CG";C
1780 Z = X + Y - 13:B = Z + 256
1790 PRINT "%YD,RX";Z;","RY";Z;","RZ";Z;","SC";S
1800 C = C + 1: IF C = 3 THEN G = 5
1810 IF C = 7 THEN C = 1
1820 PRINT "%RX";Z;"T";B;"S16"
1830 IF PEEK ( - 16384) > 128 THEN 1890
1840 PRINT "%RY";Z;"T";B;"S16"
1845 IF PEEK ( - 16384) > 128 THEN 1890
1850 PRINT "%RZ";Z;"T";B;"S16"
1855 IF PEEK ( - 16384) > 128 THEN 1890
1860 PRINT "%"
1870 NEXT X: NEXT Y
1880 W = W - 1: IF W > 0 THEN 1750
1890 POKE - 16368,0: RETURN
```

Appendix E: GRAFBASIC Program BASIC Listing

```
10 GOTO 110
20 VTAB 1: PRINT :F = 1:A = 0
30 A = A + 1: IF A = 3 THEN A = 5
40 IF A > 7 THEN A = 1
50 Z = Y:Y = X:X = A
60 GOSUB 200
70 C = PEEK ( - 16384): IF C < 128 THEN 30
80 NORMAL : PRINT "%CT3":F = 0: GOTO 100
90 C = PEEK ( - 16384): IF C < 128 THEN 90
100 POKE - 16368,0:C$ = CHR$ (C - 128): RETURN
110 IF PEEK (25472) = 32 THEN 140
120 PRINT "BLOADOBJ.SUPERG"
130 POKE 28672,255
140 I = 16
150 CALL 25472
160 PRINT "%ND"
170 PRINT "%SH28672"
180 VTAB 1: PRINT
185 PRINT "%SC?":DS = PEEK (6)
190 PRINT "%NW,PX64,PY-48 ,SC8,YD"
200 HOME : INVERSE
210 PRINT "%CT";Z
220 PRINT "
"
230 PRINT "%CT";Y
240 PRINT " << SUPERGRAPHICS >> (C) P. LUTUS 1980
"
250 PRINT "%CT";X
260 PRINT "
"
270 IF F THEN RETURN
280 PRINT "%CT3"
290 NORMAL : PRINT
300 PRINT "OPTIONS:": PRINT
310 PRINT "(S)HOW: (R)OT (T)RAN (P)OS (S)CALE"
320 PRINT "      : OF (X),(Y),(Z). "
330 PRINT "      : (C)OMMAND (*)ZERO"
340 PRINT "      : (-) CHANGE DIRECTION"
350 PRINT "      : (I)NCREMENT SIZE (Q)UIT"
360 IF F THEN RETURN
370 PRINT : PRINT
```

```
380 PRINT "(D)RAW IMAGE": PRINT
390 PRINT "(F)ILE SAVE OR LOAD"
400 PRINT : PRINT : PRINT "PRESENT IMAGE ---->"
410 VTAB 23
420 PRINT "SELECT (LETTER) : "; CHR$(8);
430 GOSUB 440: GOTO 180
440 GOSUB 20: HOME : PRINT
450 IF C$ = "?" THEN RETURN
460 IF C$ = "Q" THEN PRINT "%TE": END
470 IF C$ = "D" THEN 770
480 IF C$ = "S" THEN 520
490 IF C$ = "I" THEN 1130
500 IF C$ = "F" THEN 1160
510 RETURN
520 PRINT "%NW,PX,PY": HOME : F = 1: GOSUB 310: F =
0
530 GOSUB 90: PRINT : PRINT "%NW": GOTO 550
540 HOME : GOSUB 90
550 IF C$ = "Q" THEN RETURN
560 IF C$ = "I" THEN GOSUB 1130: GOTO 520
570 IF C$ < > "C" THEN 620
580 HOME : PRINT "ENTER SUPERGRAPHICS      "
590 PRINT "COMMAND : "; CHR$(8);: INPUT "";C$
600 PRINT "%";C$
610 HOME : PRINT "
": GOTO 540
620 IF C$ < > "*" THEN 650
630 PRINT "%NW,PX,PY,RX,RY,RZ, TX, TY, TZ, SC8"
640 GOTO 540
650 IF C$ < > "R" AND C$ < > "T" AND C$ < > "S"
AND C$ < > "P" THEN RETURN
660 L$ = C$
670 HOME : PRINT "BASIC COMMAND : ";L$;" (X,Y,Z)?
"
680 GOSUB 90
690 IF C$ = "-" THEN I = - I: GOTO 680
700 M$ = L$ + C$
710 IF C$ < > "X" AND C$ < > "Y" AND C$ < > "Z"
THEN GOTO 550
720 PRINT "%";M$;"?":Q = PEEK (6) + I
730 PRINT "%";M$;Q
740 IF Q > 127 THEN Q = Q - 256
750 HOME : PRINT "BASIC COMMAND : ";M$;" = ";Q;"
"
```

```
760 GOTO 680
770 FOR AD = 28672 TO 38400 STEP 4
780 IF PEEK (AD) < > 255 THEN NEXT AD
782 X = 0:Y = 0:Z = 0:M$ = ""
790 PRINT "%NW,PX,PY48,SC";DS
800 P = AD - 24: IF P < 28672 THEN P = 28672
810 VTAB 15
820 FOR D = P TO AD - 4 STEP 4
830 B1 = PEEK (D)
840 IF B1 = 255 THEN 940
850 L$ = "M": IF B1 > 127 THEN L$ = "D":B1 = B1 -
128
860 C$ = ""
870 IF B1 < > 0 THEN C$ = CHR$ (B1 + 48)
880 L$ = L$ + C$
890 A = PEEK (D + 1):B = PEEK (D + 2):C = PEEK
(D + 3)
900 A = A - 256 * (A > 127):B = B - 256 * (B >
127):C = C - 256 * (C > 127)
920 PRINT L$;: HTAB 5: PRINT "X = ";A;: HTAB 15:
PRINT "Y = ";B;: HTAB 25: PRINT "Z = ";C
930 NEXT D
940 VTAB 21: PRINT "(D)RAW,(M)OVE,(1-7) COLOR, (-)
DELETE"
950 PRINT "EXAMPLES: 'D3','M'. ENTER : "; CHR$
(8);: INPUT "";L$
955 IF L$ = "" OR L$ = "Q" THEN RETURN
960 IF L$ < > "-" THEN 1000
970 AD = AD - 4: IF AD < 28672 THEN AD = 28672
980 POKE AD,255: GOTO 790
1000 Q$ = LEFT$ (L$,1)
1010 IF Q$ < > "M" AND Q$ < > "D" THEN 940
1020 Q = 128 * ( MID$ (L$,1,1) = "D")
1030 IF LEN (L$) < 2 THEN 1050
1040 Q = Q + ASC ( MID$ (L$,2,1)) - 48
1050 VTAB 21: HTAB 1
1060 PRINT "
";
1070 VTAB 21: HTAB 1
1080 PRINT "'L$'" "M$" X : " CHR$ (8);: INPUT
"";X$
1082 PRINT "'L$'" "M$" Y : " CHR$ (8);: INPUT
"";Y$
1084 PRINT "'L$'" "M$" Z : " CHR$ (8);: INPUT
```

```
"";Z$
1085 IF X$ < > "" THEN X = VAL (X$):X = X + 256
* (X < 0)
1086 IF Y$ < > "" THEN Y = VAL (Y$):Y = Y + 256
* (Y < 0)
1088 IF Z$ < > "" THEN Z = VAL (Z$):Z = Z + 256
* (Z < 0)
1090 M$ = "(NO ENTRY=LAST VALUE)"
1100 POKE AD,Q: POKE AD + 1,X: POKE AD + 2,Y: POKE
AD + 3,Z
1110 AD = AD + 4: POKE AD,255
1120 GOTO 790
1130 HOME : PRINT "%NW,PX,PY"
1140 PRINT "ENTER INCREMENT SIZE (NOW ";I;") : ";
CHR$ (8);
1150 INPUT "";I: RETURN
1160 HOME : PRINT "%NW,PX,PY"
1170 FOR AD = 28672 TO 38400 STEP 4
1180 IF PEEK (AD) < > 255 THEN NEXT AD
1190 PRINT "FILE SIZE : ";AD - 28672;" BYTES":
PRINT
1200 PRINT "ENTER FILE NAME : "; CHR$ (8);
1205 INPUT "";L$
1210 IF L$ = "" THEN RETURN
1220 PRINT
1230 PRINT "(L)OAD OR (S)AVE : "; CHR$ (8);: INPUT
"";Q$
1240 IF Q$ = "" THEN RETURN
1250 IF Q$ = "L" THEN Q$ = "LOAD"
1260 IF Q$ = "S" THEN Q$ = "SAVE"
1270 PRINT "B";Q$;L$;" ,A28672";
1280 IF Q$ = "SAVE" THEN PRINT " ,L";AD - 28670
1290 PRINT : RETURN
```

Appendix F: Translate

Grafbasic will only work with Applesoft in ROM. If you have Appleworld, Supergraphics' sister product also from United Software of America, and Integer BASIC in ROM, then you can use TRANSLATE. Grafbasic is not a full text editor. Although easier to use than PEEKING and POKING it doesn't have full capabilities. If you have Appleworld it does have a full texteditor. TRANSLATE will take Appleword FIL. files and change them into files compatible with thr "SH" command.

Restrictions:

The Appleworld file must include only:
Comments (;)
Commands (N A)
Colors

and X,Y,Z inputs.

all values must be in the proper range
(-127 to 127)

Directions:

Load and Run TRANSLATE.

1). Prompt FILE ? will appear. Type in the Appleworld file name. Do not include the FIL prefix, it is included for you automatically. If you get an "FILE NOT FOUND" error and the file is in your catalog then probably you put in a space when you saved the file and your not putting in a space now.

2). Prompt SHAPE ADDRESS ? will appear. Type in a Decimal (Base 10) number indicating where you want the shape table to go. Make sure that the area of memory is not used by Supergraphics. Allowable rangw is -32767 to 32767.

3). The shape table will be placed in memory. A prompt will tell you the starting and ending addresses and the length.

4.) To continue, Run again.

Appendix G

Creating images:

This is a sample on how to make supergraphics shape tables. We will make the pyramid (SH293889) and the Spaceship at the beginning of the demo (SH 28672).

The first thing to do is graph out the figure on a sheet of paper. (see figure a.) Determine the coordinates. For the pyramid the four points are:

X	Y	Z
-1	-1	0
0	1	1
0	1	-1
1	-1	0

Determinet the easiest path that will cover all of the lines you wish to draw. Try to make as many of the lines continuations, rather than redrawing new lines. for the pyramid you need 2 sets of lines.

Set one -1,-1,0 to 0,1,1 to 0,1,-1 to -1,-1,0 to 1,-1,0 to 0,1,1

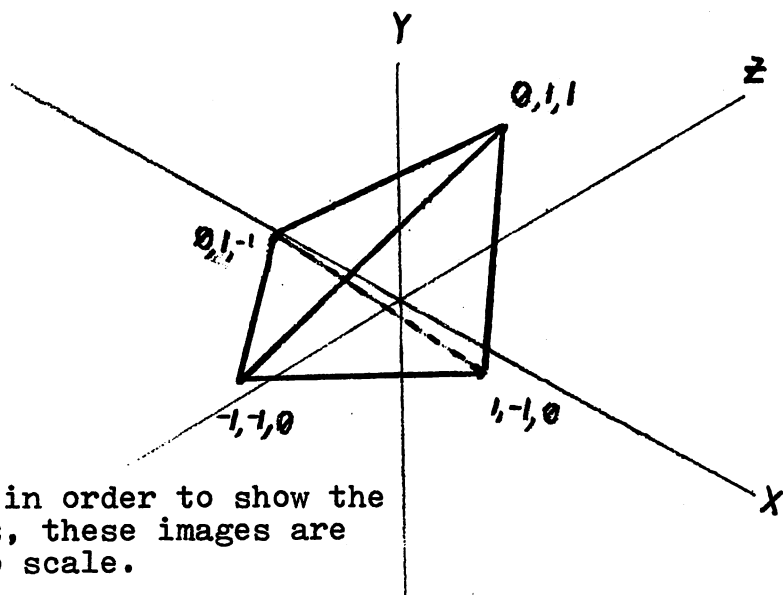
Set two 0,1,-1 to 1,-1,0

Set up the following table:

Move/color	X	Y	Z
0	-1	-1	0
	0	1	1
	0	1	-1
	-1	-1	0
	1	-1	0
	0	1	1
0	0	1	-1
	1	-1	0

and POKE the table in to RAM.

Note: on the master disk each of the points has been multiplied by 127 for a better image (see note after Spaceship).



Note: in order to show the points, these images are not to scale.

Figure A. the pyramid.

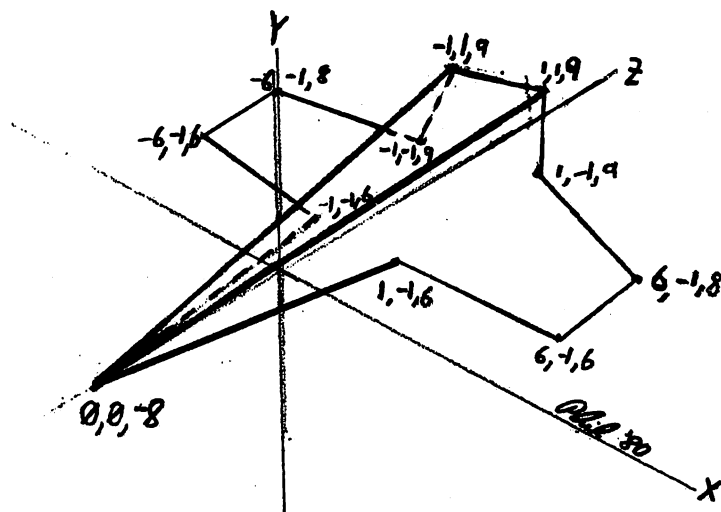


Figure b. The Spaceship

The Spaceship.

For the Spaceship there are eleven points.
they are:

X	Y	Z
0	0	8
1	-1	6
6	-1	6
6	-1	8
1	-1	9
1	1	9
-1	1	9
-1	-1	9
-6	-1	8
-6	-1	6
-1	-1	6

Believe it or not this image, although more complex can also be done with two lines.

They are

	X	Y	Z
from	0	0	-8
to	-1	-1	6
to	-6	-1	6
to	-6	-1	-8
to	-1	-1	9
to	-1	1	9
to	1	1	9
to	1	-1	9
to	6	-1	8
to	6	-1	6
to	1	-1	-6
to	0	0	-8
to	-1	1	9
and			
from	0	0	-8
to	1	1	9

In order to make the image finer quality, multiply each point by eight, making the image larger and finer in quality (for large images especially it will make better shapes).

Appendix H

The following addresses are given for those programmers who wish to program Supergraphics directly in 6502 machine language. The information is given as-is. The source listings are proprietary and will not be given out.

HEX	LABEL	DESCRIPTION
638D	INIT	start program
63E5	ERASE	erase screen
6444	READER	handles characters on screen
64AA	READ3	execute complete ASCII command in buffer \$6F0E to first <CR>
66BE	PRCHAR	output ASCII to both HIRES
662D	DOTXT	same as %TE
63C8	DOGRF	same as %GR
6622	QUIT	same as %QT
6637	GETDIG	reads numeric ASCII from buffer
65DF	CONLOOP	executes assembly language loop and checks keyboard
6869	DRAWIT	draws figure (what did you expect)
6554	YD	same as %YD
65AB	SC	sets scale to accumulator (%SC)
65BB	CT	sets color (%CT)
6922	PLOT	plots point at oldX,oldy
6952	LINE	plots from oldX,oldY to plotX, plotY
67E9	SCROLL	scrolls window
6C19	SETCOL	sets up color masks for color in A call this before using PLOT,LINE

Variable table

Locations	Variables
351	XROT
352	YROT
353	ZROT
354	X TRAN
355	Y TRAN
356	Z TRAN
357	X POS
358	Y POS
359	SCAL X
35A	SCAL Y
35B	SCAL Z
35C	GRAPHICS COLOR
35D	SHAPE ADDRESS LO
35E	SHAPE ADDRESS HI
08	OLDX
09	OLDY
06	PLOTX
07	PLOTY
33D	NUMBER LO
33E	NUMBER HI
33F	LOOPFROM LO
340	LOOPFROM HI
341	LOOPTO LO
342	LOOPTO HI
343	LOOPSTEP
334	SCBAS (\$20 for screen 0) (\$40 for screen 1)
336	BUFPTR Points to end of command buffer
2D7	BUFPOS points to current chr in buffer
333	SCREEN 0 or 1 for HIRES Screen
332	PLOTFLAG 0 = no draw 1 = yes draw
2D6	TEXTFLAG 0 = graphics 1 = text
33D	LASTIN last chr received from COUT
33C	COMMAND current command

COMMAND LIST

0	RX
1	RY
2	RZ
3	TX
4	TY
5	TZ
6	PX
7	PY
8	SX
9	SY
A	SZ
B	CG
C	SH
D	TE
E	GR
F	NW
10	QT
11	SC
12	CT
13	YD
14	ND

<end>

